the line is freed), or updated in a lazy manner, when a remote PE requests a line which is no longer present (at a transaction cost penalty for the requesting PE when the line is requested).

DL 9-14-07

Please replace the paragraph beginning on Page 15, line 14 with the following:

For example, a penalty can occur when P$ 326 sub-lines are requested from remote PEs, but are never used by the requesting PE. This can occur due to false sharing on a CDRAM line basis, and given the coarse nature of the CDRAM line may be common for many applications that have not been optimized for this environment. Updates to the coherence directory may be frequent due to request and release of successive memory units. The first issue can be solved by providing the P$ 326 with those sub-lines that are available and need not be requested using coherence actions. Thus, prefetch is only performed on available lines, reducing the cost of false sharing. In an optimized embodiment, predictors might be used to determine if certain sub-lines, which are not available, should be requested, while identifying other sub-lines, which are not subject to requests, by the P$ 326 file mechanism.

DL 9-14-07

Please replace the paragraph beginning on Page 16, line 27 with the following:

CD 405 also receives/sends indicia through a bus PB 410 at the same inferred target rate, to external I/O (BIF) ports. Resident and transient data is processed independently by PEs, PE 422, PE 432, PE 442 and PE 452, each with its own bi-directional CDRAM to Chip Stacked Ports ([[CBIF]]CPIF), CPIF 420, CPIF 430, CPIF 440 and CPIF 450, all operating at an inferred rate of ¼ terabyte [that is, 250 gigabytes] per second (A terabyte is a measure of computer storage capacity and is 2 to the 40th power or approximately a thousand billion bytes - that is, a thousand gigabytes).

Each PE device is of approximately 70mm2 dimensions and uses 0.10 micron SOI [silicon-on-insulator] Process Technology (CMOS 10S) [a proprietary technology developed by IBM].

DL 9-14-07

Please replace the paragraph beginning on Page 22, line 7 with the following:

In one embodiment, this includes using a (possibly slower) coherence bus provided[[in a]]. In another preferred embodiment, this involves operations performed in a dedicated state machine, microcode, millicode, firmware, or software. In one embodiment, the coherence bus is a physical multi-drop coherence bus, in a further embodiment, the coherence bus is a logical coherence bus composed of multiple point-to-point links. In yet another embodiment there is provided a coherence ring. In other embodiments, coherence actions use special signaling primitives between nodes provided by the system (such as, including but not limited to, processor to processor interrupts), or a messaging and signaling system provided in the memory. Eventually, coherence actions obtain the data with the required access mode and provide them to at least one processor core in the node for either processing or storage with the requested access mode.

DL 9-14-07

Please replace the paragraph beginning on Page 23, line 21 with the following:

In step 860, there is performed a third test, to see if the data can be successfully acquired in exclusive mode, that is,[[,]] the directory information does not indicate any use in shared or exclusive mode. If so, step 862 indicates success. This is possible because acquiring data items in exclusive mode is compatible with memory-resident state only, but not when data are maintained in any state in another node.

DL 9-14-07

Please replace the paragraph beginning on Page 24, line 4 with the following:

In a first step 870, the shared and exclusive conditions are determined. (Specific bits in this format are used to perform efficient coherence actions, but are not necessary for the execution of step 820.) In step 872, a first test is performed to determine if the directory information has a legal form. If not, control transfers to step 874 which invokes special condition handling mechanisms implemented in conjunction with FIGURE 8, such as error handling, and protocol retry. In step 876, a second test is performed to test if the data items are required in shared mode, and are not in exclusive mode. If so, success is indicated in step 878. In step 880, there is performed a third test, to see if the data can be successfully acquired in exclusive mode, that is,[[,]] the directory information does not indicate any use in shared or exclusive mode. If so, step 882 indicates success. Otherwise, coherence actions are necessary, and this is indicated in step 884. The statements associated with these calls are as follows.

DL 9-14-07

Please replace the paragraph beginning on Page 25, line 13 with the following:

wherein the subscription operator [I] indicates the bit numbered I of a bit vector, the operator = indicates assignment, and the operator OR corresponds to the logical OR of two Boolean values. The variables request.exclusive and request.shared indicate whether the request was for shared or exclusive access mode. [[(]]In another embodiment, the request is indicated by a single bit.[[).]]

DL 9-14-07

Please replace the paragraph beginning on Page 26, line 11 with the following:

wherein the subscription operator [I] indicates the bit numbered I of a bit vector, and the operator = indicates assignment. The variables request.exclusive and request.shared indicate whether the request was for shared or exclusive access mode, and request.node indicates the number of the requesting node. [[(]]In another embodiment, the request is indicated by a single bit.[[)]]

DL 9-14-07

Please replace the paragraph beginning on Page 27, line 6 with the following:

There is now set forth another exemplary embodiment of the preferred embodiment, according to the instruction set of the preferred embodiment. According to the exemplary embodiment, there are [[at]] four nodes connected to a shared memory hierarchy level via point-to-point links, the links providing means for requesting data for immediate processing or local storage in a cache for future processing. Each node consists of at least one processor. In one embodiment, each node is a heterogeneous system, consisting of at least one processing unit and one auxiliary processing unit. In another embodiment, there is provided a heterogeneous SMP, wherein at least one node has at least one first processing unit, and at least one node has at least one second processing unit (such as an auxiliary processing unit). According to this embodiment, there is supported prefetching capability. In one embodiment, a page containing a data item is prefetched. In another embodiment, other sequences of data items are prefetched. For making the features of this invention apparent, we will describe the embodiments in the context of prefetching a page surrounding the specific data item request.

DL 9-14-07

Please replace the paragraph beginning on Page 30, line 9 with the following:

In step 1230, when it is determined that the requested data has been successfully retrieved from shared memory hierarchy level 1000 with permissions [[which]] <u>that</u> are compatible with the requested access mode, [[and]] the data is provided in step 1230 to at least one processor core in the node for either processing or storage with the requested access mode. In step 1235, the received page and associated directory information is optionally stored in a prefetch page cache.

DL 9-14-07

Please replace the paragraph beginning on Page 32, line 4 with the following:

Referring now to step 1225, a method in accordance with FIGURES 8A and 8B can be used, employing a directory for the specific memory subunit being requested. In another embodiment, the methods of FIGURES 8A and 8B are extended to include prefetch information in the testing steps. In one embodiment, this is achieved by extending steps 850 and 870, respectively, to incorporate the prefetch address information into the tests being performed on state variables "shared" and "exclusive". In another embodiment, such steps incorporating prefetch information in directory information is performed by logic in the shared memory hierarchy level 1000.

DL 9-14-07

Please replace the paragraph beginning on Page 34, line 19 with the following:

Turning now to FIGURE 13, there are shown exemplary methods implemented by the shared memory hierarchy level 1000. In a first method 1305, the shared memory hierarchy level 600 receives a request in a first step 1310,[[,]] for example, in accordance with the exemplary FIGURE 11, over a point to point link from a node. In a second step 1315, a page containing the requested data item as well as other data items hierarchically composing a page selected from data items 1005-1012 are returned to the requesting node, in conjunction with the directory information state 1015-1022 associated with those data items being transferred, as well as contents of prefetch address registers. In one embodiment, all prefetch registers are transmitted. In another embodiment, only those registers having a possible conflict are indicated. In another embodiment, the contents of the prefetch registers 1030-1036 are combined with the directory information 1015-1022 before being transferred in step 1315. In another step [[920]] 1320, directory information is updated. In another step 1325, page cache information such as prefetch address registers 1030-1036 are updated. Steps 1315, 1320 and 1325 are preferably implemented atomically with respect to other

protocol transactions, in particular to other execution instances of steps 1315-1320, or 1340-1345 on behalf of other nodes.

DL 9-14-07

Please replace the paragraph beginning on Page 35, line 18 with the following:

It will also be apparent that according to the embodiments set forth in this invention, the memory is used as a central integration point. In one aspect of integration, coherence is performed not by a central directory controller, or through a snoop of a central bus, but by reading [[and]] an updated directory information of a memory component in the system.

9-14-07

Please replace the paragraph beginning on Page 36, line 1 with the following:

In another aspect of this invention, chip stacking is used to physically integrate the system, processing element chips being physically stacked onto or with a memory component, and the memory component serves as physical integration point. In one specific embodiment, C4 connectors are used to connect chip-stacked processing elements and memory component chips. In another aspect of this invention, wire-bonding is used to connect chip-stacked processing elements and memory component chips.

DL 9-14-07

Please delete the paragraph beginning on Page 37, line 3, which begins with, "It is understood that the present invention can take many forms and implementations. . ."

DL 9-14-07

Please delete the paragraph beginning on Page 37, line 13, which begins with, "Having thus described the present invention by reference to certain of its salient characteristics. . ."

9-10-07

SPECIFICATION

PL 9-14-07

Please amend the Specification as follows:

Please replace the paragraph beginning on Page 12, line 31 with the following:

PE 210 is attached to MUX 215 via BUS 212. ~~PE2 30~~ PE 230 is attached to MUX 235 via BUS 232. PE 250 is attached to MUX 255 via BUS 252 and PE 270 is attached to MUX 275 via BUS 272. Similarly, in one exemplary embodiment, external memory is optionally connected to the system via an I/O processor (not shown, and not limited to a single main memory or single I/O processor), via BUS 207 to MUX 205. Likewise, in one exemplary embodiment, BUS 283 optionally conducts signals between MUX 285 and a visualizer or other I/O devices (not shown for clarity). (A visualizer is the interface to a graphics display device.)[[.]]